

[19]中华人民共和国国家知识产权局

[51]Int. Cl⁶

H04Q 11/04

[12] 发明专利申请公开说明书

[21] 申请号 97195425.9

[43]公开日 1999年6月30日

[11]公开号 CN 1221548A

[22]申请日 97.4.3 [21]申请号 97195425.9

[30]优先权

[32]96.4.10 [33]US[31]08/630,578

[86]国际申请 PCT/SE97/00570 97.4.3

[87]国际公布 WO97/38550 英 97.10.16

[85]进入国家阶段日期 98.12.10

[71]申请人 艾利森电话股份有限公司

地址 瑞典斯德哥尔摩

[72]发明人 L·G·W·埃尼罗斯

L·G·佩特森

[74]专利代理机构 中国专利代理(香港)有限公司

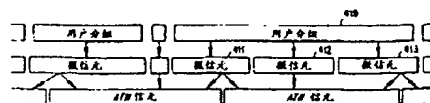
代理人 程天正 李亚非

权利要求书 5 页 说明书 9 页 附图页数 10 页

[54]发明名称 微信元分段及重组

[57]摘要

在使用以 ATM 自适应层(AALm)协议作为数据传输基础设施的异步转移模式(ATM)的电信系统中,一种将用户数据分组分段和重组的方法和设备。该方法和设备通过避免与过长微信元有关的问题而改善电信数据传输的质量和效率。



ISSN 1008-4274

权 利 要 求 书

1. 在电信系统中, 一种产生数据信元的方法包括如下步骤:
将数据分组分成至少两段;
将至少两段中的每一个插入各个微信元; 以及
5 将相应微信元复接成至少一个数据信元.
2. 权利要求 1 的方法, 其特征在于将数据分组分段的步骤还包括如下步骤:
如果数据分组比预定长度长, 就将该数据分组分段.
3. 权利要求 2 的方法, 其特征在于预定长度与至少一个数据信元
10 的有效负荷部分等长.
4. 权利要求 1 的方法, 其特征在于将至少两个分段中的每一个插入各个微信元的步骤还包括如下步骤:
将一个码附加到至少两个分段中的每一个上.
5. 权利要求 4 的方法, 其特征在于附加到至少两个分段中每个上
15 的码定义了相应分段的长度和分段类型, 而且其中分段类型从包括第一分段类型、第二分段类型以及最后分段类型的一组分段类型中选择.
6. 权利要求 4 的方法, 其特征在于附加到至少两个分段中每个上的码定义了相应分段的长度和分段类型, 而且其中分段类型从包括第一分段类型和最后分段类型的一组分段类型中选择.
- 20 7. 权利要求 1 的方法, 其特征在于至少一个数据信元是异步转移模式信元.
8. 在电信系统中, 传输数据分组的方法包括如下步骤:
将数据分组分成至少两个分段;
将至少两个分段插入各个微信元;
25 将各个微信元复接成至少一个数据信元; 以及
将至少一个数据信元从发送实体发送.
9. 权利要求 8 的方法, 其特征在于将数据分组分段的步骤还包括如下步骤:
如果数据分组比预定长度长, 则将数据分成至少两个分段.
- 30 10. 权利要求 9 的方法, 其特征在于预定长度与至少一个数据信元的有效负荷部分等长.
11. 权利要求 8 的方法, 还包括如下步骤:

在接收实体接收至少一个数据信元;

从至少一个数据信元中提取各个微信元的每一个;

从各个微信元中提取至少两个分段; 以及

通过合并至少两个所提取的分段将数据分组重组。

- 5 12. 权利要求 11 的方法, 其特征在于将数据分组重组的步骤还包括如下步骤:

存储一个预定的重组法则集合; 以及

- 10 根据预定的重组法则集合将至少两个分段重新合并, 其中重新合并至少两个分段的步骤持续到重新合并了至少两个分段中的最后一个为止。

13. 权利要求 11 的方法, 其特征在于将至少两个分段插入各个微信元的步骤还包括如下步骤:

将一个码附加到至少两个分段的每个上。

- 15 14. 权利要求 13 的方法, 其特征在于附加到至少两个分段中每个上的码定义了相应分段的长度和分段类型, 而且其中分段类型从包括第一分段类型、第二分段类型、以及最后分段类型的一组分段类型中选择。

- 20 15. 权利要求 13 的方法, 其特征在于附加到至少两个分段中每个上的码定义了相应分段的长度和分段类型, 而且其中分段类型从包括第一分段类型和最后分段类型的一组分段类型中选择。

16. 一种产生数据信元的装置包括:

将数据分组分成至少两个分段的装置;

将至少两个分段中每个插入各个微信元中的装置; 以及

将各个微信元复接成至少一个数据信元的装置。

- 25 17. 权利要求 16 的装置, 其特征在于将数据分组分段的装置还包括:

如果数据分组比预定长度长就将该数据分组分成至少两段的装置。

18. 权利要求 17 的装置, 其特征在于预定长度与至少一个数据信元的有效负荷部分等长。

- 30 19. 权利要求 16 的装置, 其特征在于将至少两个分段中的每一个插入各个微信元的装置还包括:

将一个码附加到至少两个分段中的每一个上的装置。

20. 权利要求 19 的装置, 其特征在于将一个码附加到至少两个分段中每个上的装置还包括:

用于附加码的装置, 该码定义了相应分段的长度和分段类型, 而且其中分段类型从包括第一分段类型、第二分段类型以及最后分段类型的一组分段类型中选择。

21. 权利要求 19 的装置, 其特征在于将一个码附加到至少两个分段中每个上的装置还包括:

附加码的装置, 该码定义了相应分段的长度和分段类型, 而且其中分段类型从包括第一分段类型和最后分段类型的一组分段类型中选择。

22. 权利要求 16 的装置, 其特征在于至少一个数据信元是异步转移模式信元。

23. 传输数据分组的装置包括:

将数据分组分成至少两个分段的装置;
将至少两个分段插入各个微信元的装置;
将各个微信元复接成至少一个数据信元的装置; 以及
将至少一个数据信元从发送实体发送的装置。

24. 权利要求 23 的装置, 其特征在于将数据分组分成至少两个分段的装置还包括:

如果数据分组比预定长度长则将数据分组分成至少两个分段的装置。

25. 权利要求 24 的装置, 其特征在于预定长度与至少一个数据信元的有效负荷部分等长。

26. 权利要求 23 的装置, 还包括:

在接收实体接收至少一个数据信元的装置;
从至少一个数据信元中解复接各个微信元的每一个的装置;
从它们各自的微信元中提取至少两个分段的装置; 以及
通过合并至少两个所提取的分段将数据分组重组的装置。

27. 权利要求 26 的装置, 其特征在于通过合并至少两个所提取的分段将数据分组重组的装置包括:

存储一个预定的重组法则集合的装置; 以及
根据预定的重组法则集合将至少两个分段重新合并, 直到重新合并

了最后分段为止的复接装置。

28. 权利要求 26 的装置, 其特征在于将至少两个分段插入各个微信元的装置还包括:

将一个码附加到至少两个分段的每个上的装置。

5 29. 权利要求 28 的装置, 其特征在于该码定义了相应分段的长度和分段类型, 而且其中分段类型从包括第一分段类型、第二分段类型、以及最后分段类型的一组分段类型中选择。

30. 权利要求 28 的装置, 其特征在于该码定义了相应分段的长度和分段类型, 而且其中分段类型从包括第一分段类型和最后分段类型
10 的一组分段类型中选择。

31. 产生数据信元的电信系统包括:

一个控制逻辑电路, 将数据分组分成至少两个分段;

一个第一复接器, 将至少两个分段中的每一个插入各个微信元; 以
及

15 一个第二复接器, 将各个微信元插入至少一个数据信元。

32. 权利要求 31 的电信系统, 其特征在于, 如果数据分组长度比预定长度长, 则控制逻辑电路将数据分组分段。

33. 权利要求 32 的电信系统, 其特征在于预定长度与至少一个数据信元的有效负荷部分等长。

20 34. 权利要求 31 的电信系统, 其特征在于还包括:

存储至少两个码的连接表, 其中, 第一复接器将至少两个码分别附加到至少两个分段上。

35. 权利要求 34 的电信系统, 其特征在于, 至少两个码中的每一个定义了相应分段的长度和分段类型, 而且其中分段类型从包括第一
25 分段类型、第二分段类型以及最后分段类型的一组分段类型中选择。

36. 权利要求 34 的电信系统, 其特征在于, 至少两个码中的每一个定义了相应分段的长度和分段类型, 而且其中分段类型从包括第一分段类型和最后分段类型的一组分段类型中选择。

37. 权利要求 31 的电信系统, 其特征在于至少一个数据信元是异步
30 转移模式信元。

38. 传输数据分组的电信系统包括:

一个控制逻辑电路, 将数据分组分成至少两个分段;

一个第一复接器，将至少两个分段中的每一个插入各个微信元；以及

一个第二复接器，将各个微信元插入至少一个数据信元；以及
一个发射机，将至少一个数据信元从发送实体发送。

5 39. 权利要求 38 的电信系统，其特征在于，如果数据分组比预定长度长，则控制逻辑电路将数据分组分成至少两个分段。

40. 权利要求 39 的电信系统，其特征在于预定长度与至少一个数据信元的有效负荷部分等长。

41. 权利要求 38 的电信系统，还包括：

10 一个第一数据缓存器，在接收实体接收至少一个数据信元；

一个解复接器，从至少一个数据信元中提取各微信元中的每一个；

一个第二解复接器，从它们各自的微信元中提取至少两个分段；以及

及

15 一个第二控制逻辑电路，通过重新合并至少两个所提取的分段来控制数据分组的重组。

42. 权利要求 41 的电信系统，还包括：

一个连接表，存储一个预定的重组法则集合，

其中第二控制逻辑电路根据预定的重组法则集合将至少两个分段重新合并，直到重新合并了最后分段为止。

20 43. 权利要求 41 的电信系统，其特征在于第一复接器将一个码附加到至少两个分段的每个上。

44. 权利要求 43 的电信系统，其特征在于，至少两个码的每一个定义了相应分段的长度和分段类型，而且其中分段类型从包括第一分段类型、第二分段类型、以及最后分段类型的一组分段类型中选择。

25 45. 权利要求 43 的电信系统，其特征在于，至少两个码的每一个定义了相应分段的长度和分段类型，而且其中分段类型从包括第一分段类型和最后分段类型的一组分段类型中选择。

说明书

微信元分段及重组

背景

5 本发明涉及电信数据的传输，更具体地涉及使用异步转移模式（ATM）协议的电信数据传输。特别是，本发明涉及一种用于将数据传输分组分成更小的分组以便改善数据传输的效率的方法和设备。

ATM 是一种在电信系统（例如，蜂窝电信系统）中发送电信数据的标准协议。它基于以固定大小信元（称为 ATM 信元）传输数据，这里
10 每个 ATM 信元有 48 个八比特的有效负荷以及 5 个八比特组（octet）字头。ATM 在本领域中是熟知的，用于低比特率应用是很常见的（例如，蜂窝话音通信）。但是，ATM 不能有效地利用低比特率应用中的带宽。

带宽是非常昂贵的；因此，使带宽的利用率最大化是非常重要的。当将 ATM 用于低比特率通信时，带宽利用率可以通过结合图 1 所示的 ATM
15 自适应层（AALm）100 来改善。一般，AALm 建立在将用户数据（例如，话音数据）压缩成很小的称为微信元的数据分组的基础上。AALm 可以再划分三个子层：会聚子层 101、打包拆包（AAD）子层 102、以及复接解复接（MAD）子层 103。会聚子层 101 用做电信应用（即，蜂窝电话系统）与 AAD 子层 102 之间的接口。AAD 子层 102 在发送实体（例如，
20 蜂窝电信系统基站）将用户数据插入微信元，并在接收实体（例如，蜂窝电信系统移动交换中心）将用户数据从微信元中提出。MAD 子层 103 在发送实体将微信元复接成 ATM 信元，并在接收实体将微信元解复接。

图 2 说明了已知方法如何使用 AALm 100 将每个用户数据分组（例如，用户分组 201）插入单个微信元（例如，微信元 202）。换句话说，
25 在每个用户数据分组和每个微信元之间存在一对一的关系。因此，每个微信元的长度可以从只有几个八比特组变化到几百个八比特组，这依据相应用户分组的长度而定。实际上，微信元可以比几个 ATM 信元（例如，微信元 203）要长。

尽管与 AALm 一起使用 ATM 可以比没有 AALm 的 ATM 较好地实现带
30 宽利用率，但是过长微信元会引起其它问题，例如带数据部分的微信元比预定长度（例如，ATM 信元有效负荷长度）要长。首先，大的微信元引入较大的延迟变化。延迟变化是指数据传输和达到时间中的变化。

延迟变化一般表现为电信信号中的“抖动”。为了避免抖动，系统必须对固定延迟加入延迟变化因子，产生较长的总传输时间。尽管对固定延迟加入延迟变化因子会降低抖动，但是较大的延迟需要使用昂贵的回声抵消器，而且它们也导致话音质量的整体下降。特别是，诸如
5 话音通信这样的低比特率应用，十分依赖于稳定的数据传输延迟（即，较小的延迟变化）；因此，低比特率应用特别容易受到前面提到的、由于用户数据在过长微信元中传输所引起的退化效应的影响。

与使用长微信元有关的第二个问题是当电信网络或端设备将微信元从一个 ATM 流交换到另一个时产生的。如果微信元用户分组比 ATM 信
10 元有效负荷短或相等，那么在交换机入端将微信元放入 ATM 信元、将该微信元在所需方向上进行交换、在出端提取出微信元、以及将微信元复接成新的 ATM 流则几乎没有什么问题。

简单地说，过长微信元，特别是，带有比 ATM 信元有效负荷长的数据部分的微信元可能降低语音质量以及网络交换设备的效率，因此就
15 产生必须通过对用户数据分组分段从而限制长微信元的需要。

发明概要

本发明的一个目的是提供一种有效地利用可用带宽的电信数据传输协议。

发明的另一个目的是提供一种电信数据传输协议，它有效地利用可
20 用带宽并减小与通过过长微信元传输电信数据有关的语音质量问题。

发明的另一个目的是提供一种电信数据传输协议，它有效地利用可用带宽并避免与将过长微信元从一个 ATM 流交换到另一个有关的问题。

根据本发明的一个方面，前述以及其它目的是通过一种用于产生数据信元的方法、装置、或一种电信系统实现的，其中包括：将数据分
25 组至少分成两段，将至少两段中的每一段插入各个微信元；并将各个微信元复接成至少一个数据信元。

根据发明的另一方面，一种用于传输数据分组的方法、设备或电信系统包括：将数据分组分成至少两段；将该至少两段插入各个微信元；将各个微信元复接成至少一个数据信元；并将该至少一个数据信元从
30 发送实体进行发送。

附图的简要描述

发明的目的和优点将通过结合附图阅读如下详细描述而理解，其

中:

图 1 说明了一种已知的 AALm 协议模型;

图 2 表示一种用于将用户数据分组插入微信元的已知方法;

图 3 说明带新的分段和重组子层的 AALm 协议模型;

5 图 4a 和 4b 表示根据新的分段和重组子层将用户数据分组插入微信元的方法;

图 5 描述了根据“三码方法 (three code method)”的分段过程;

图 6 表示代表根据“三码方法”的重组过程的状态图;

图 7 说明根据“三码方法”的微信元字头码;

10 图 8 描述了根据“三码方法”的分段过程;

图 9 表示代表根据“三码方法”的重组过程的状态图;

图 10 说明根据“三码方法”的微信元字头码;

图 11 说明将用户数据分组分段并组成微信元的设备; 以及

图 12 说明将微信元拆包并对用户数据重组的设备。

15 详细描述

本发明将过长用户数据分组分段并将每段插入一个小微信元中。相反, 已知方法是将整个用户数据分组在一个很长的微信元中发送。通过将用户数据分组分段并在较小的微信元中发送用户数据分组, 与较长微信元有关的问题, 例如语音质量下降以及网络交换设备失效, 都可以减小或消除。

20 本发明通过对 AALm 协议模型引入一个新的功能子层来实现这些功能。图 3 描述了带有新功能子层 301 的 AALm 300。新功能子层 301 称为分段和重组 (SAR) 子层。如果用户数据分组太长以至必须分段以避免在长度 (除了字头) 超过预定最大长度 (例如, ATM 有效负荷长度) 的微信元中将用户数据发送到接收实体, SAR 子层 301 就被激活。

图 4a 表示了发送实体 401 (例如, 蜂窝电信基站)、互连链路 402、以及接收实体 403 (例如, 移动交换中心)。发送实体 401 以及接收实体 403 都包含图 3 中说明的新协议模型。特别是, 发送实体 401 包含 SAR 子层 301 的分段部分, 接收实体 403 包含 SAR 子层 301 的重组部分。

30 互连链路 402 将 ATM 信元从发送实体 401 传输到接收实体 403, ATM 信元再将分段的用户数据 (例如话音通信信号) 在微信元中传输。

图 4b 说明了本发明的新 AALm 协议模型是如何取得每个长用户分

组（例如，用户分组 410）、对其进行分段、并将它放置到很多小的微信元（例如微信元 411、412、和 413）中。不象已知的 ATM 协议模型（参考图 2），每个用户数据分组和每个微信元之间不再存在一对一的对应。特别是，图 4b 说明了与图 2 所示的已知协议模型相比较，单个
5 微信元不再覆盖一个以上的 ATM 信元边界。这是因为，如上面所讨论的，每个微信元的长度是有限的，例如，不超过 ATM 信元有效负荷的长度（即 48 个八比特组）。

有两种基本方法实现根据本发明的分段和重组。这两种方法中的一种都不意味着使本发明限于这两种方法。相反，这两种方法被认为
10 反映了发明的两种特定实施例。第一种方法或实施例称为“三码方法”。第二种方法或实施例称为“二码方法”。一般而言，两种实施例使用同一基本分段策略。用户分组被分成几个分段。除了最后分段以外的所有分段都有固定并且相等的长度。最后分段的长度是可调节的，从而使所有分段的总长等于原始用户分组的长度。然后将各分段放入微
15 信元有效负荷。因此，每个微信元有效负荷的长度与每个相应的用户分组分段的长度相等。

图 5 表示发送实体 401 在将数据传输到接收实体 403 之前完成的“三码方法”或实施例的分段过程 500。例如，假设用户分组 501 长度为 178 个八比特组。为了说明的目的，固定分段大小设为 16 个八比特
20 组。但是，本领域的技术人员将会理解，固定分段大小可以设为任意所需的大小。因此，带有 16 个八比特组长有效负荷的 11 个微信元（例如，微信元 502 和 503）以及带有两个八比特组长有效负荷的 1 个微信元（例如，微信元 504）将分段的用户分组 501 从发送实体 401 传输到接收实体 403。

25 在发送实体 401，SAR 子层 301 调用 AAD 子层 302。AAD 子层 302 将微信元字头附加到每个微信元上（例如，微信元字头 505、506 及 507）。除了其它以外，微信元字头规定了相应的有效负荷的长度以及微信元是否对应于“第一分段”302、“中间分段”503、或“最后分段”504。

30 在接收实体 403，AAD 子层 302 提取微信元字头，该字头通知 AAD 子层 302：该微信元究竟对应于“第一分段”、“中间分段”、或“最后分段”。AAD 子层 302 继续将分段传递到 SAR 子层 301，SAR 子层将

分段逐一重组回到原始的数据分组。在 SAR 子层 301 将“最后分段”加到数据分组之后，它将重组的数据分组传递到会聚层 304。

如果数据分组的长度太短，因而不需要“中间分段”，那么三码实施例仅将数据分组分成“第一分段”和“最后分段”。如果数据分组很短，以致可以放入单个微信元，就不需要分段了。在这种情况下，发送实体 401 在标为“最后分段”的单个微信元中将数据分组发送到接收实体 403。

图 6 表示了一个状态图，代表三码实施例的重组过程 600。状态图包括三个状态：空闲态 601、重组态 602、以及退出态 603。在开始 602 或上电时，重组过程 600 进入空闲态 601。空闲态 601 只表示目前没有进行重组。

在一般过程中，只要接收实体 403 接收到标为“第一分段”的微信元——如事件 604 所示，重组过程 600 就进入重组状态 602。然后 SAR 子层 301 将与“第一分段”有关的用户数据存储起来。当接收实体 403 接收所有“中间分段”时——如事件 605 所示，重组过程 600 保留在重组状态 602。当每个“中间分段”到达时，SAR 子层 301 通过将与此些中间分段有关的用户数据按顺序加到与“第一分段”有关的用户数据上而将数据分组重组。当接收实体 403 接收到“最后分段”时，如事件 606 所示，SAR 子层 301 将相应的用户数据加到前面存储的用户数据上，然后将完全组好的用户数据分组 501 提供给下一个子层，例如，会聚子层 304。然后重组过程 600 进入空闲态 601，如事件 607 所示。

当整个数据分组可以包含在单个微信元中时，如上所述，不需要调用 SAR 子层 301。因此，接收实体 403 不进行重组。为了完整性，图 6 表示在空闲态 601 接收到“最后分段”将使重组过程 600 进入重组状态 602，如事件 613 所示。AAD 子层 302 从微信元中提取用户数据并直接提供给 SAR 子层 301 之上的子层，例如，会聚子层 304。此后，重组过程 600 重入空闲态 601。

如果遇到一个或多个特定错误，重组过程 600 将进入退出态 603。例如，在发明的另一个实施例中，可以定义表示数据分组最大长度的门限。如果 SAR 子层 301 在重组数据分组时超过这个最大长度，重组过程 600 将进入退出态 603，以便清除该错误，如事件 608 所示。据此，重组过程 600 重进入空闲态 601，如事件 609 所示。

在本发明的另一个实施例中，将规定超时值。在重组数据分组时如果 SAR 子层 301 超过超时值，则重组过程 600 将会进入退出态 603 以便清除错误，如事件 610 所示。此后，重组过程 600 重进入空闲态 601，如事件 609 所示。

- 5 当重组过程 600 处于空闲态 601 时，如果接收实体收到标为“中间分段”的微信元，重组过程 600 检测到错误并进入退出态 603 以便清除该错误，如事件 611 所示。此后，重组过程 600 重进入空闲态 601，如事件 609 所示。类似地，当处于重组态 602 时如果接收实体 403 收到标为“第一分段”的微信元，那么重组过程 600 将检测到错误并进入退出态 603 以便清除该错误，如事件 612 所示。此后，重组过程 600 重进入空闲态 601，如事件 609 所示。

图 7a、7b 和 7c 表示对微信元字头进行配置，从而可标识相应的微信元是否与“第一分段”、“中间分段”、及/或“最后分段”相关联，并标识相应分段的长度。例如，图 7a 说明四个码 48、49、50 和 51 表示微信元 701 对应于长度分别为 8、16、32 或 48 个八比特组的“第一分段”。类似地，图 7b 说明四个码 52、53、54 和 55 表示微信元 702 对应于长度分别为 8、16、32、48 个八比特组的“中间分段”。图 7c 说明码 0-47 表示微信元 703 对应于长度分别为 1-48 个八比特组的“最后分段”。

- 20 图 7a、7b 和 7c 中的特定码是说明性的。本领域技术人员将会理解可以使用其它码完成这个功能，而且如果需要的话可以分配更多或更少的码。但是，应该在发送实体 401 和接收实体 403 中预先定好特定的码值。

这种编码策略使 SAR 子层 301 按照需要将用户数据分组分段。例如，在发明的另一个实施例中，分段长度甚至可以在对应于同一用户数据分组的
25 不同段之间变化。如图 5 所示，SAR 子层 301 将用户分组 501 分成其长度为 16 个八比特组的 11 个相等的段以及长度为 2 个八比特组的一个段。但是，SAR 子层 301 也可以将用户分组分成其长度为 8 个八比特组的“第一分段”、其长度为 16 个八比特组的三个“中间分段”、
30 长度为 48 个八比特组的一个“中间分段”、其长度为 32 个八比特组的一个“中间分段”、以及其长度为 2 个八比特组的“最后分段”，总长度为 178 个八比特组。

图 7a、7b 和 7c 说明微信元字头也包含其它信息。微信元字头一般包括微信元连接标识符 (CID)。CID 将微信元连接彼此分开, 并使多个微信元连接复接成同一 ATM 连接。例如, CID 可以标识一个特定的蜂窝电话呼叫; 因此, 对应于该呼叫的数据分组将在其所具有的字头中包含相同 CID 值的微信元中传输。可以理解对应于相同分段数据分组的每个微信元包含相同的 CID 值。在本发明的另一个实施例中, 长度码 (例如图 7a 中所示的 48 到 51) 可以针对每个 CID 值来定义。因此, 每个 CID 值可以有在连接建立时定义的自己的一组固定长度码。微信元字头也包含字头完整性校验 (HIC) 码。这个码用于通过检测并纠正可能在微信元从发送实体 401 向接收实体 403 传输过程中出现的错误从而来保护字头信息。CID 和 HIC 都是本领域熟知的。

如上所述, 本发明的第二示范实施例称为“二码方法”。图 8 表示由发送实体 401 在将数据传输到接收实体 403 之前完成的“二码方法”或实施例的分段过程 800。在大多数情况下, SAR 子层 301 按照与三码实施例中相同的方式将用户数据分组 801 分段, 不同之处在于: 例如将一个合并码用于“第一分段”和所有“中间分段”, 如微信元字头 802 和 803 所示。

例如, 假设分段的用户分组 801 的长度是 178 个八比特组。然后, 为了说明起见, 固定分段大小设置为 16 个八比特组。因此, 11 个带有长度为 16 个八比特组的有效负荷的微信元 (例如, 微信元 804 和 805) 以及一个带有长度为 2 个八比特组的有效负荷的微信元 (例如, 微信元 806) 将携带用户分组 801 从发送实体 401 传送到接收实体 403。

图 9 说明了一个状态图, 表示二码实施例的重组过程 900。类似图 6 中的状态图 600, 图 9 中的状态图包括三个状态: 空闲态 901、重组态 902、以及退出态 903。由于图 9 中描述的每个事件与参照图 6 描述的具有类似标号的事件相同, 这里就不再重复了。但是, 注意图 9 也说明在重组过程 600 中引起错误检测的事件 611 和 612 将不在二码实施例中使用。

图 10a 和 10b 说明了二码实施例编码方案的例子。图 10a 特别表示微信元字头中只需要较少的码, 因为不必区分“第一分段”和“中间分段”。

图 11 说明了使用上面描述的 AALm 300 实现电信信号的分段及组

装的示范硬件实施例 1100. AAL_m 300 首先将用户分组 1101 从较高子层 (即, 会聚子层 304) 传递到 SAR 子层 301. 附加在用户分组 1101 上的是一个表示 CID 和 ATM 信元连接的指针 1102. FIFO-IN 1103 存储并确定用户分组 1101 的长度. 然后, 复接器 (MUX-IN) 1104 将指针 1102 从用户分组 1101 中提取出来并发送一个代表指针 1102 中包含的信息的控制信号 1105 到控制逻辑 1106. 控制逻辑 1106 使用这个控制信号来选择连接表 1108 中特定的地址 1107. 连接表 1108 包含用于对用户分组 1101 分段的预装载的规则, 并包含对每个用户分组 1101 的分段构造微信元字头所需的信息. 例如, 当已知用户分组 1101 的长度以及每个单个分段的长度, 则预装载的法则可以定义用户分组 1101 将要被划分的分段的特定数目. 特别是, 每个微信元连接对应于连接表 1108 中的不同地址. MUX-IN 1104 根据用户分组 1101 的长度 (由 FIFO-IN 1103 提供) 以及连接表 1108 中地址 1107 处存储的预装载法则来计算每个分段的实际大小.

15 为了组装每个微信元, 控制逻辑 1106 从连接表 1108 的地址 1107 处取得 ATM 连接和微信元字头信息. 控制逻辑 1106 将这个信息发送到复接器 (MUX-OUT) 1109, 复接器通过将微信元字头信息附加到 FIFO-IN 1103 所提供的用户分组 1101 的相应分段上来产生微信元. MUX-OUT 1109 也将恰当的 ATM 连接信息附加到每个微信元上, 如 ATM 指针 1110 所说明的. 在将微信元导向 MAD 子层 302 之前, SAR 子层 301 将微信元存储在 FIFO-OUT 1111 中.

在本发明的另一个实施例中, SAR 子层 301 可以使用多个 FIFO-IN 装置来并行地将几个用户分组分段. 此外, FIFO-IN 1103 可以与连接表 1108 共处于相同位置.

25 图 12 说明了在接收实体 403 使用上面描述的 AAL_m 300 来实现微信元拆包以及用户分组重组的示范硬件实施例 1200. 该过程开始时, MAD 子层 302 在接收实体 403 将微信元 1201 与相应的 ATM 指针 1202 一起存储在 FIFO-IN 1203 中. ATM 指针 1202 表示该微信元是从哪一个 ATM 信元解复接的.

30 然后, 控制逻辑 1204 使用 MUX-IN 1205 提取 ATM 指针 1202 以及微信元字头. 微信元字头信息包含一个如上所述的表示每个微信元是与“第一”、“中间”、或是“最后”相关联 (根据所使用的是三码

实施例还是二码实施例)的码。例如,这可以再表示一个新的重组过程是否将开始、正在进行的重组过程是否将继续、或重组过程是否结束。

5 如果微信元字头表示正在启动新的重组过程,控制逻辑 1204 就从连接表 1207 中取得指针 1206,并将它与第一微信元所关联的数据分段一起放置到 FIFO-OUT 1208 中。指针 1206 的位置(即地址)由 ATM 指针 1202 结合微信元字头中的 CID 来确定。一旦指针 1206 和第一分段存储在 FIFO-OUT 1208 中,所有来自属于同一微信元连接(即,具有相同 CID 值)的连续微信元的随后数据分段都通过使用 MUX-IN 1205
10 和 MUX-OUT 1209 而被发送到 FIFO-OUT 1208 中。当“最后分段”到达,SAR 子层 301 将全部重组的用户分组 1210 导向下一子层,例如,会聚子层 304。

如前,SAR 子层 301 可以使用多个 FIFO-OUT 装置来并行地从多个微信元连接中重组微信元。此外,FIFO-OUT 装置 1209 可以与连接表 1204
15 共处于相同位置。

本发明已经参考几个示范实施例进行了描述。但是,对于本领域技术人员很显然的是:用特定形式、而不是上面描述的示范实施例中的那些来实施本发明是可能的。这可以在不背离发明精神的前提下进行。这些示范实施例只是说明性的,而且不应该认为在任何方面有所限制。
20 本发明的范围通过所附的权利要求给出,而不是前面的描述,而且所有落入权利要求范围内的变化和等效都认为是包括于其中的。

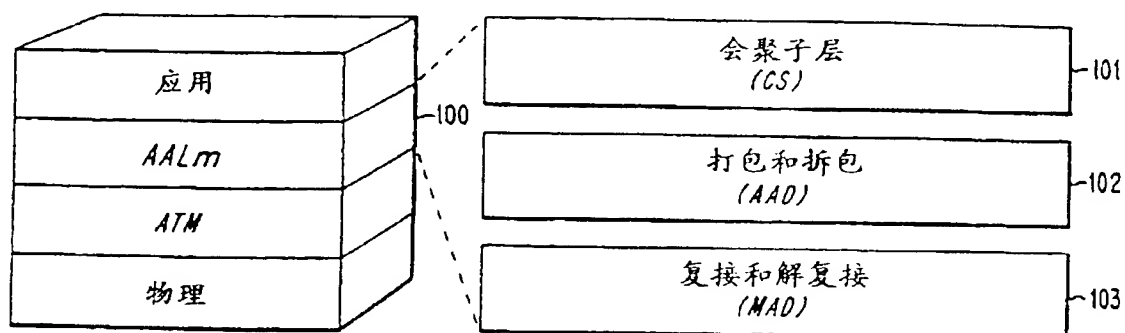


图 1
(现有技术)

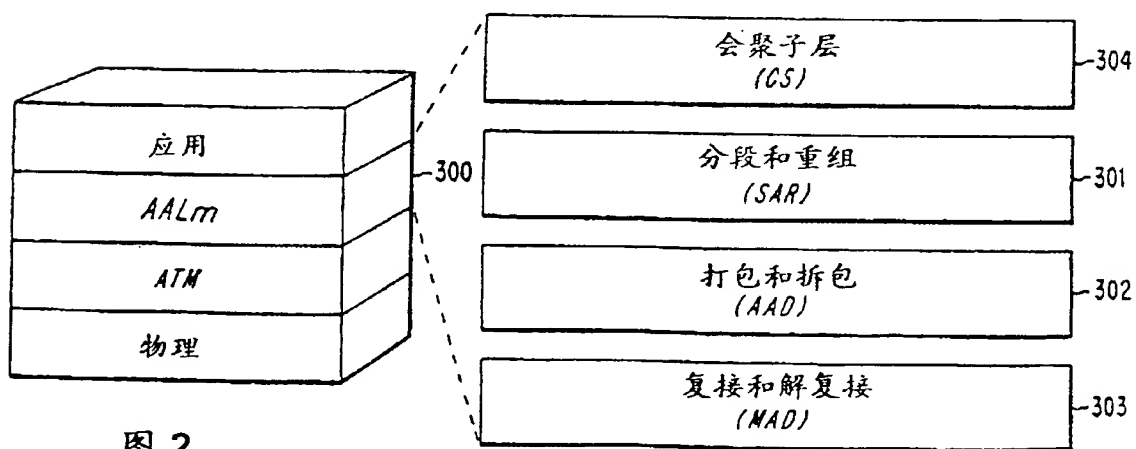


图 2

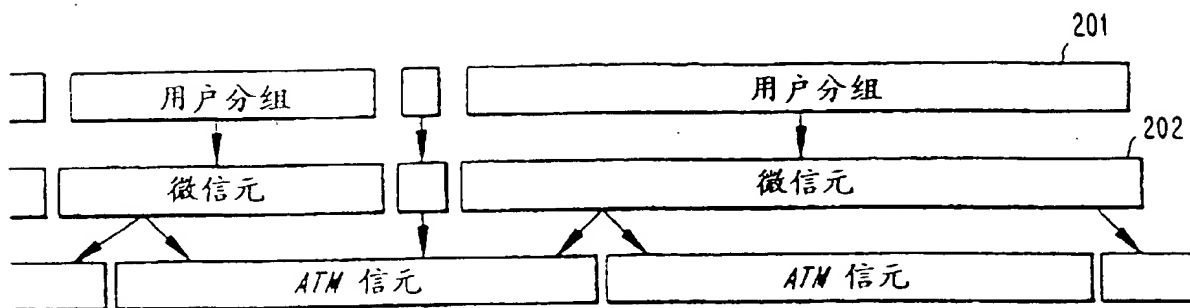


图 2
(现有技术)

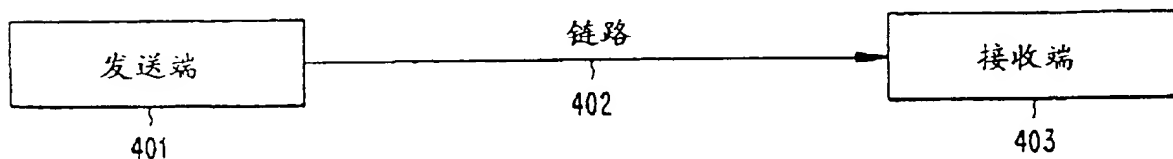


图 4a

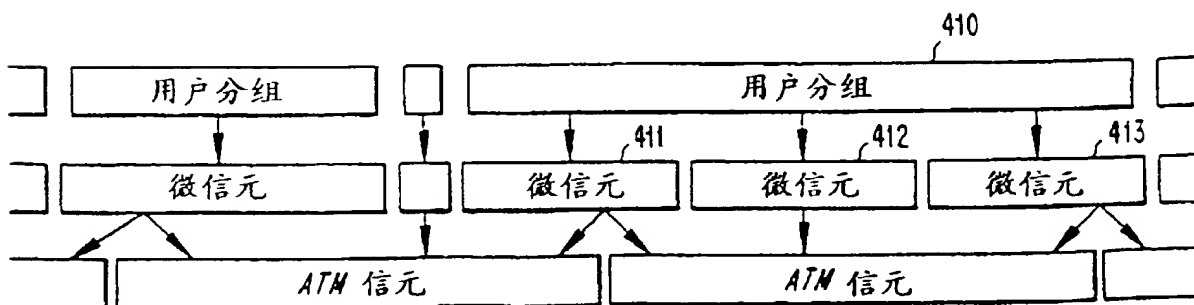
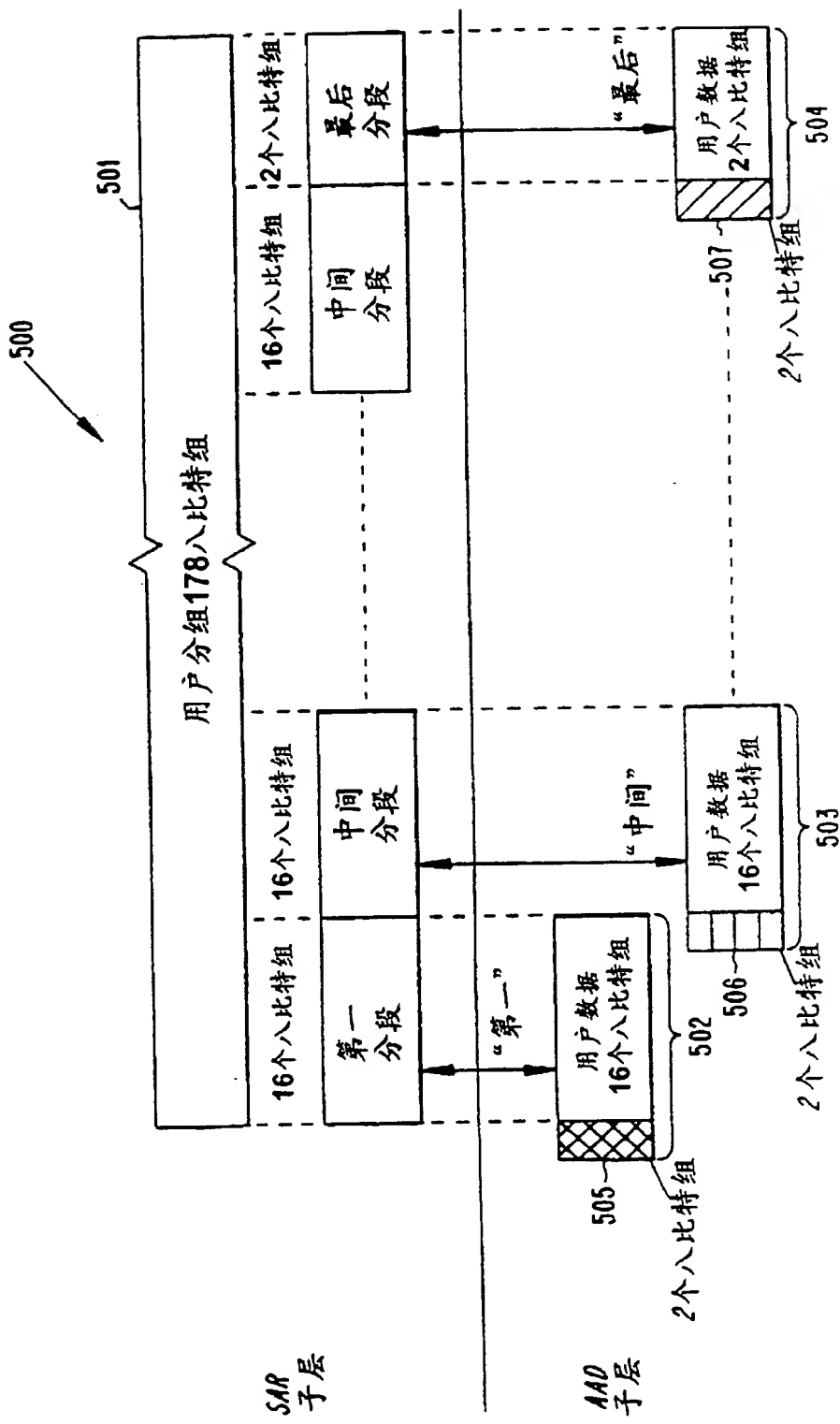


图 4b



微信元字头编码
“最后分段”

微信元字头编码
“中间分段”

微信元字头编码
“第一段”

图 5

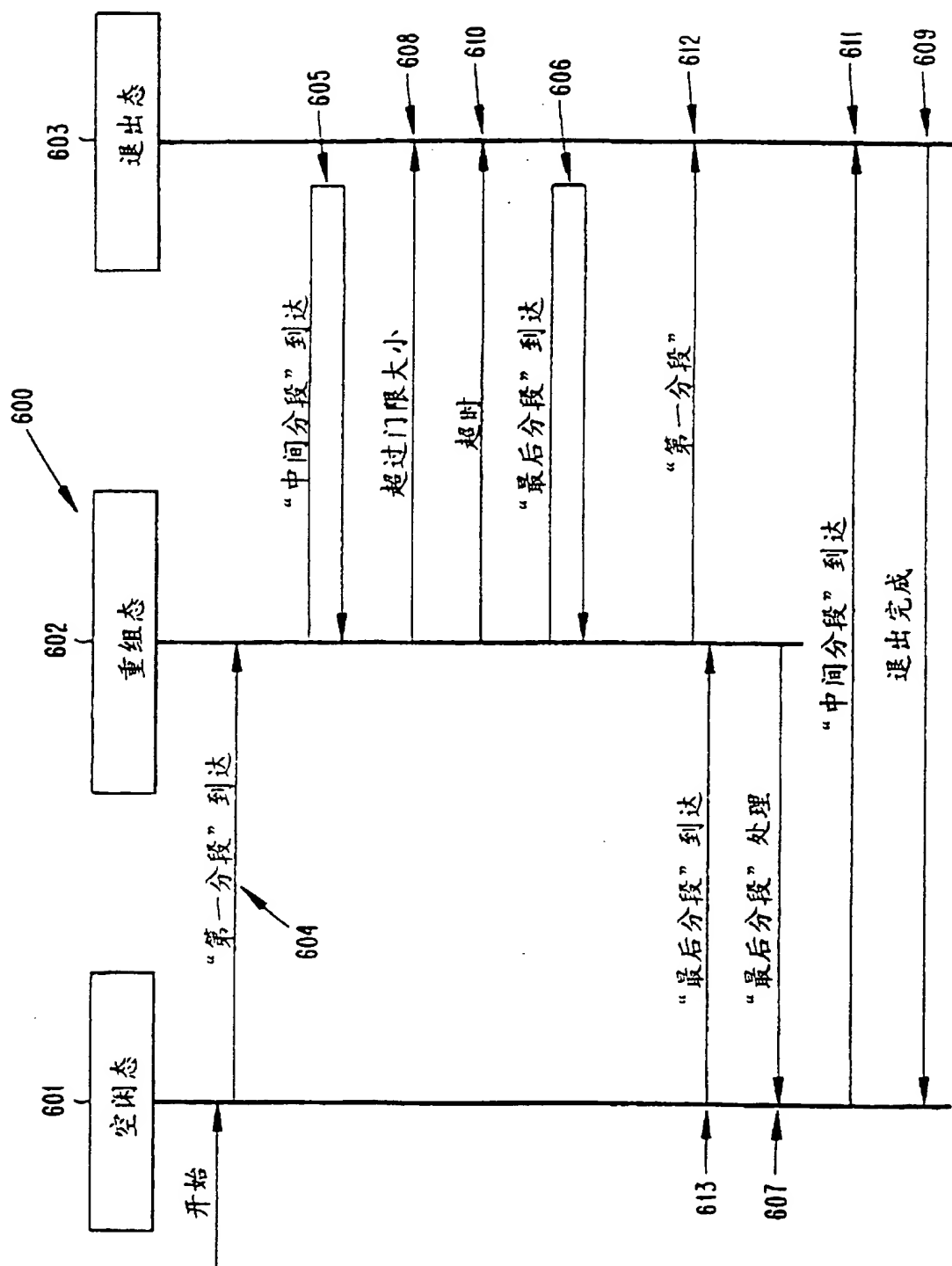


图 6

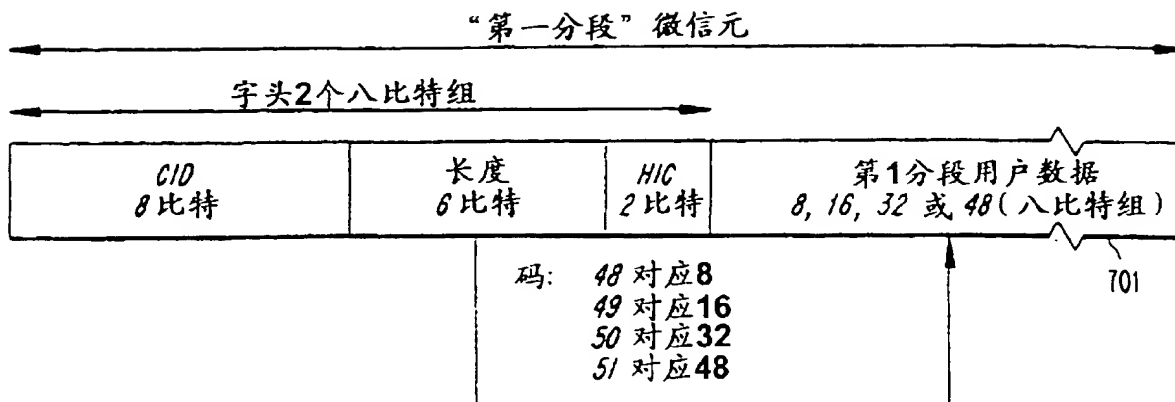


图 7a

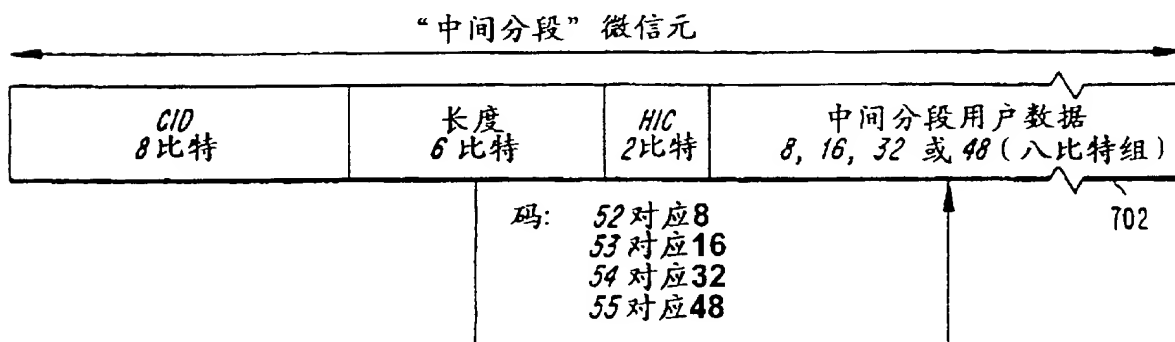


图 7b

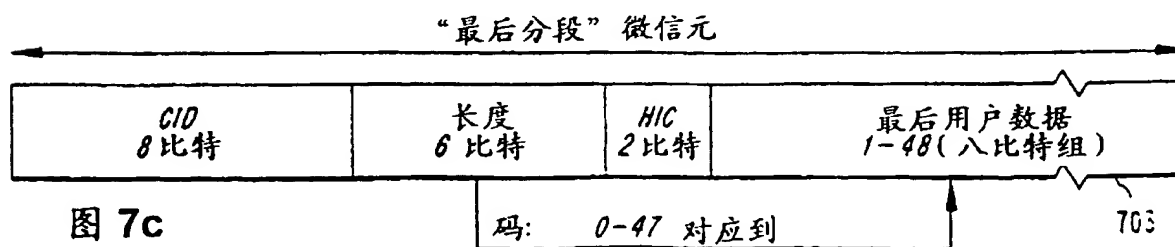


图 7c

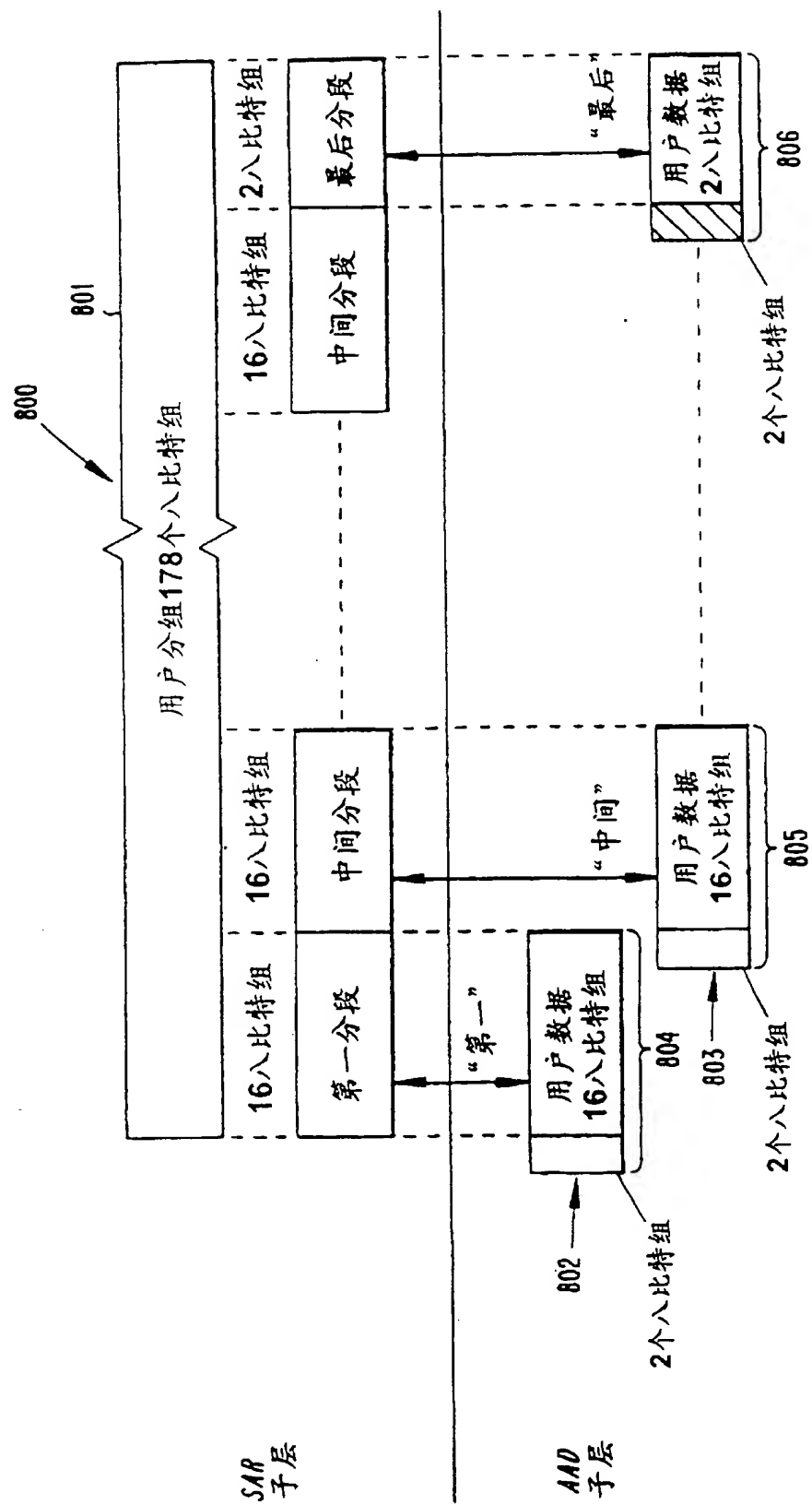


图 8

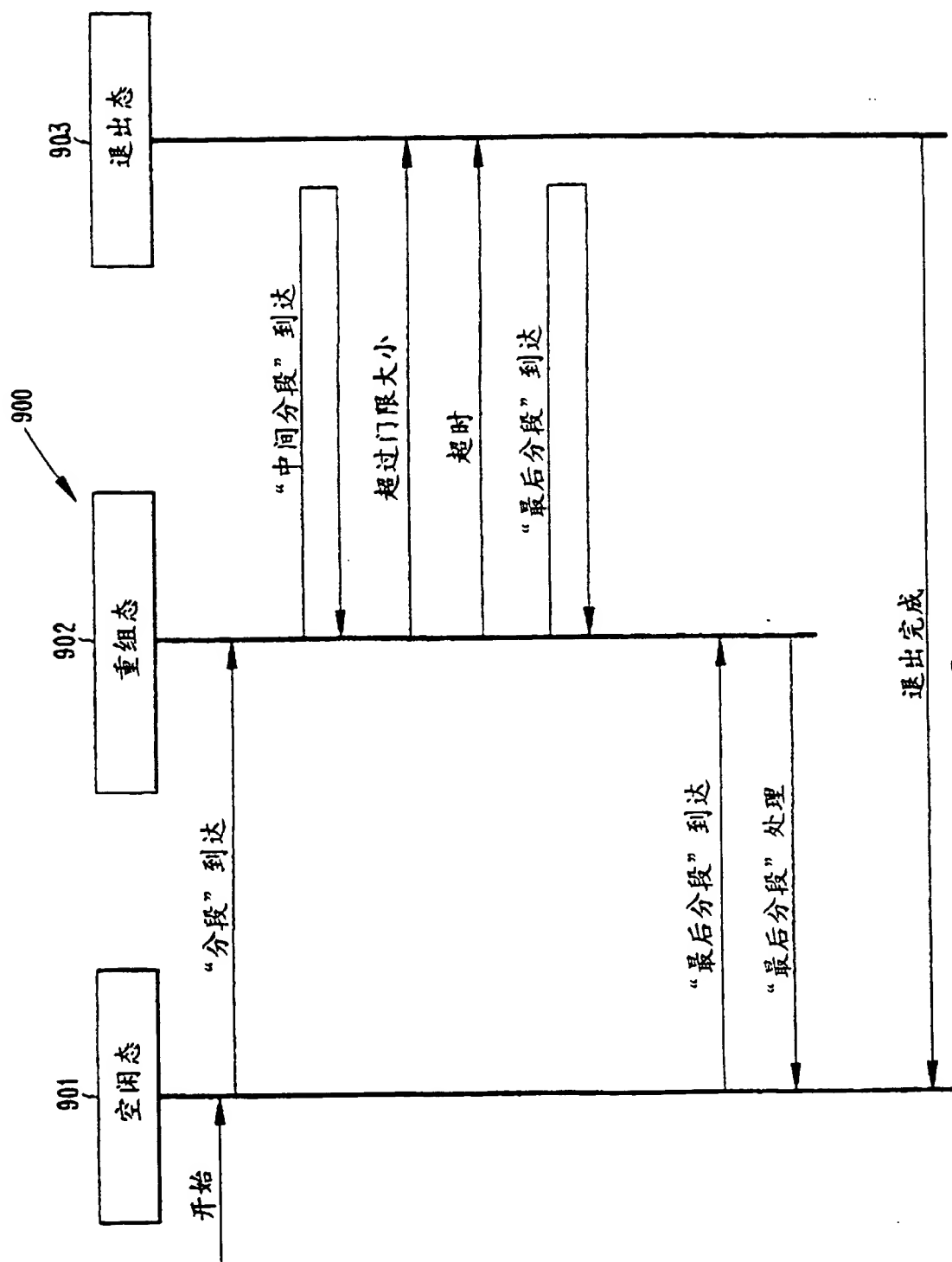


图 9

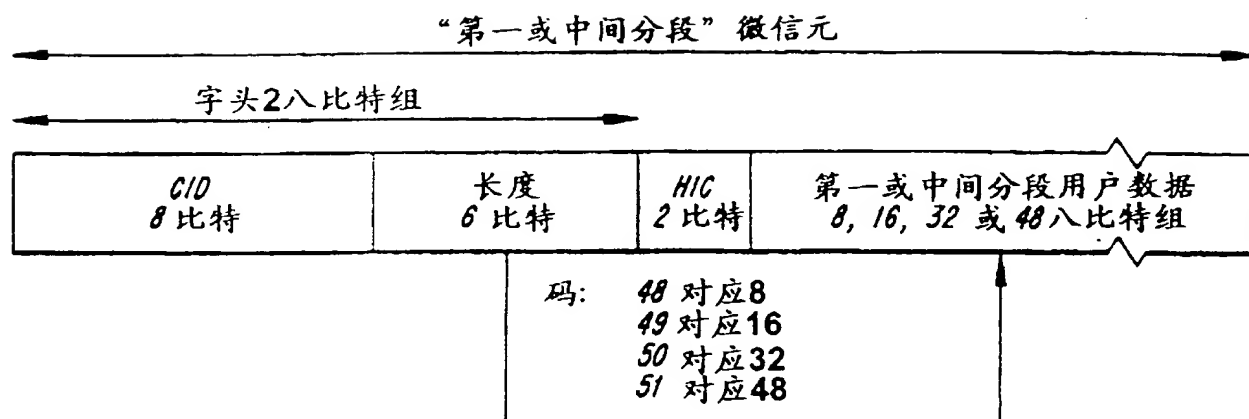


图 10a

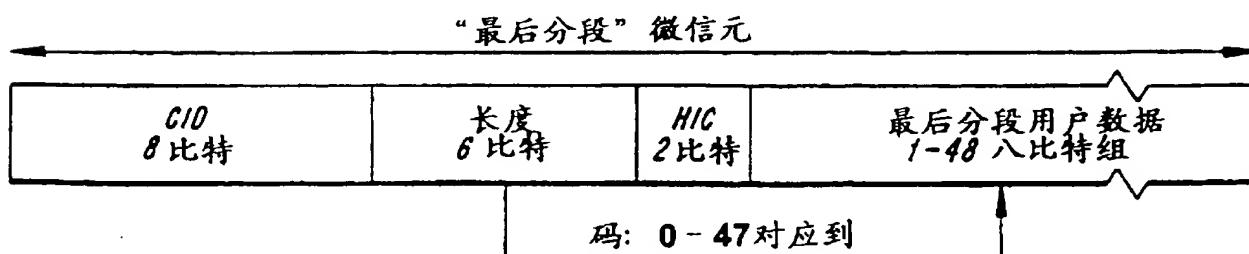


图 10b

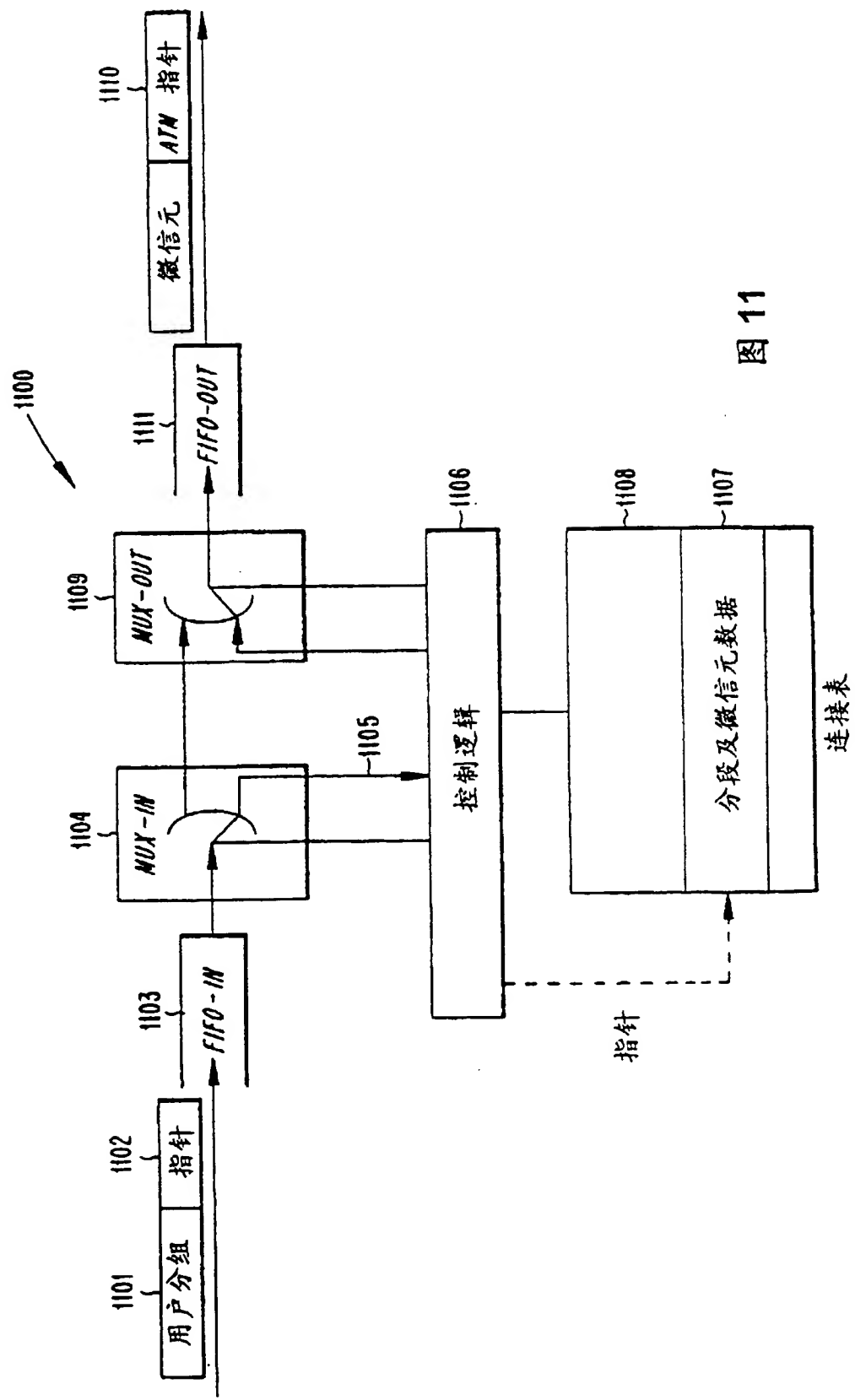


图 11

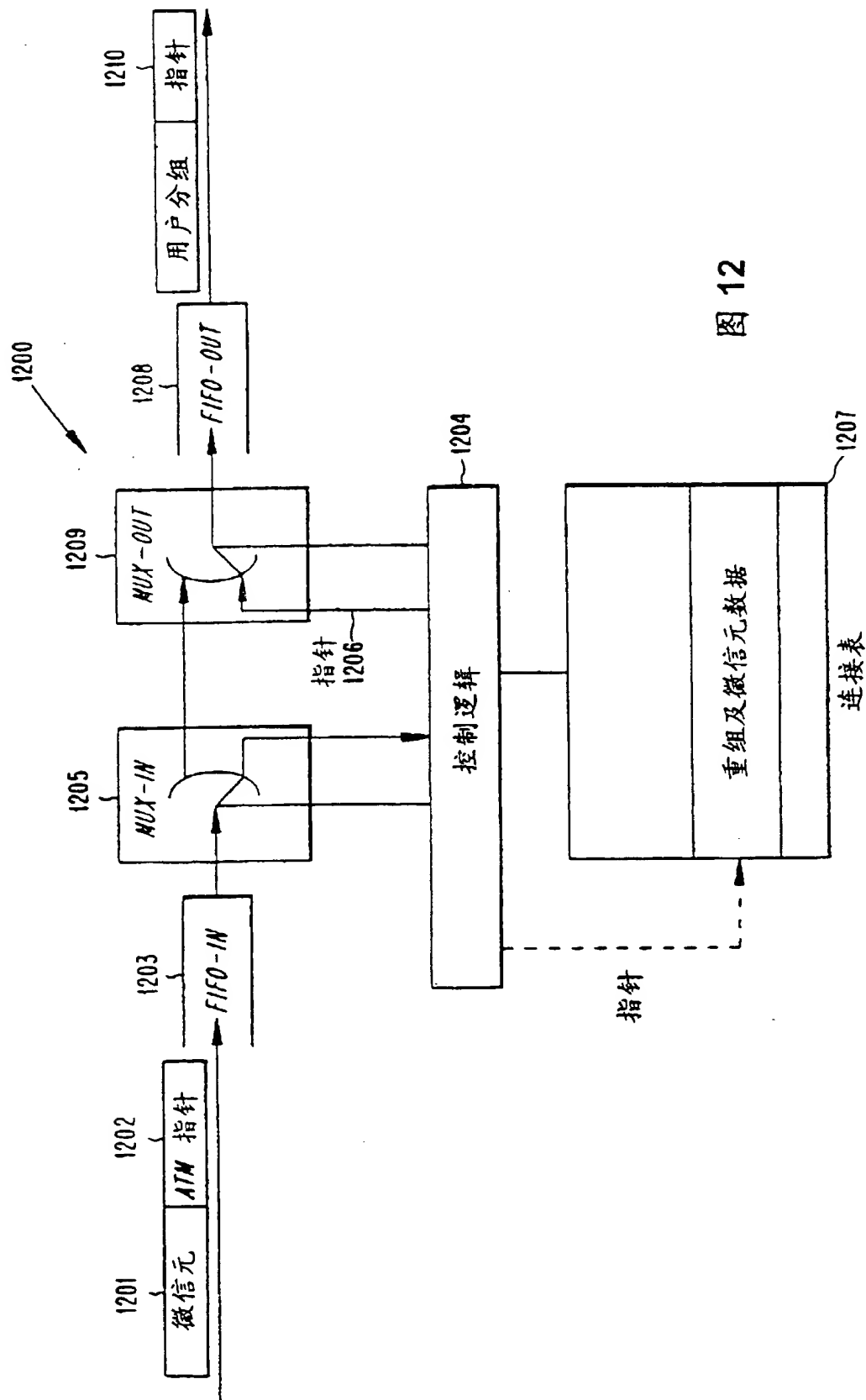


图 12

